MINI-DB
Demystifying the Inner Workings of Database Systems

Hossein Hakimzadeh, Robert Batzinger, Susan Gordon

Department of Computer and Information Sciences
Indiana University – South Bend, Indiana

Nov. 8-10, 2010
Las Vegas, NV

23rd International Conference on
Computers and Their Applications
in Industry and Engineering
(CAINE-2010)

INDIANA UNIVERSITY

MINI-DB
Demystifying the Inner Workings of Database Systems



Hossein Hakimzadeh, Robert Batzinger, Susan Gordon

Department of Computer and Information Sciences
Indiana University – South Bend, Indiana

International Society for Computers and their Applications

Nov. 8-10, 2010
Las Vegas, NV

23rd International Conference on
Computers and Their Applications in Industry and Engineering
(CAINE-2010)

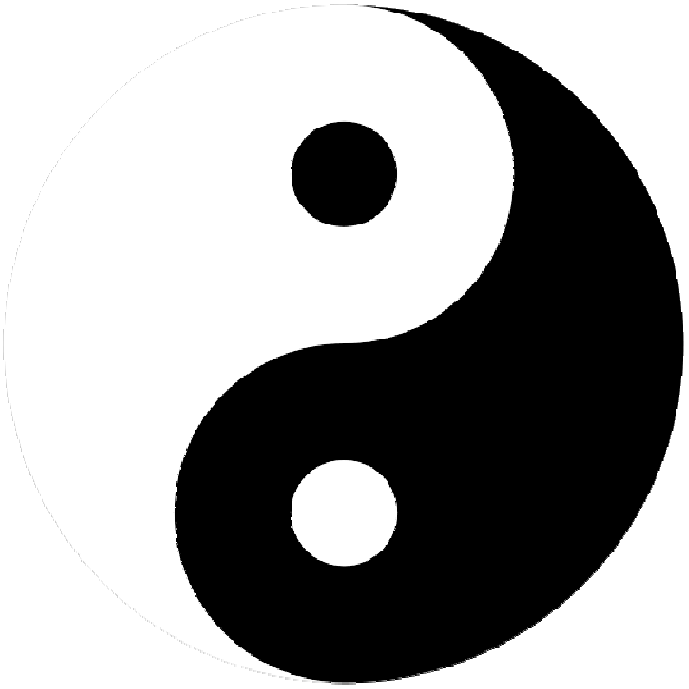
INDIANA UNIVERSITY

Outline

- The Challenge
- Our Solution!
- MINI-DB
- Lessons learned – Student Feedback
- Conclusions

The Challenge:

- Diversification of the CS Curriculum
  - Advantages
  - Disadvantages



Diversification of CS Curriculum:

- Advantages
  - Ability to expose students to contemporary topics such as cyber security, distributed computing, parallel computing, bioinformatics, and game programming, robotics, etc.

Diversification of CS Curriculum:

- Disadvantages
  - Courses that deal with the internal working of computers, or courses that require system design and system development are being systematically removed from the undergraduate curriculum.
  - Merging of (OS and Networking), (Concepts of Programming Languages and Compilers), (File Organizations and Databases)

Our Solution:

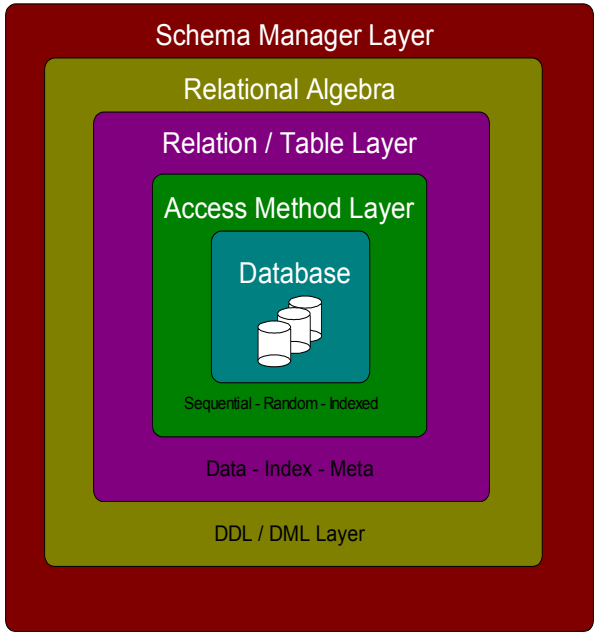
- Deliberate review and redesign of elective and required courses to include system design and development.
- Development of more project based courses.
- Development of Open Source Courseware. (e.g. http://www.ocwconsortium.org/)

Case Study:

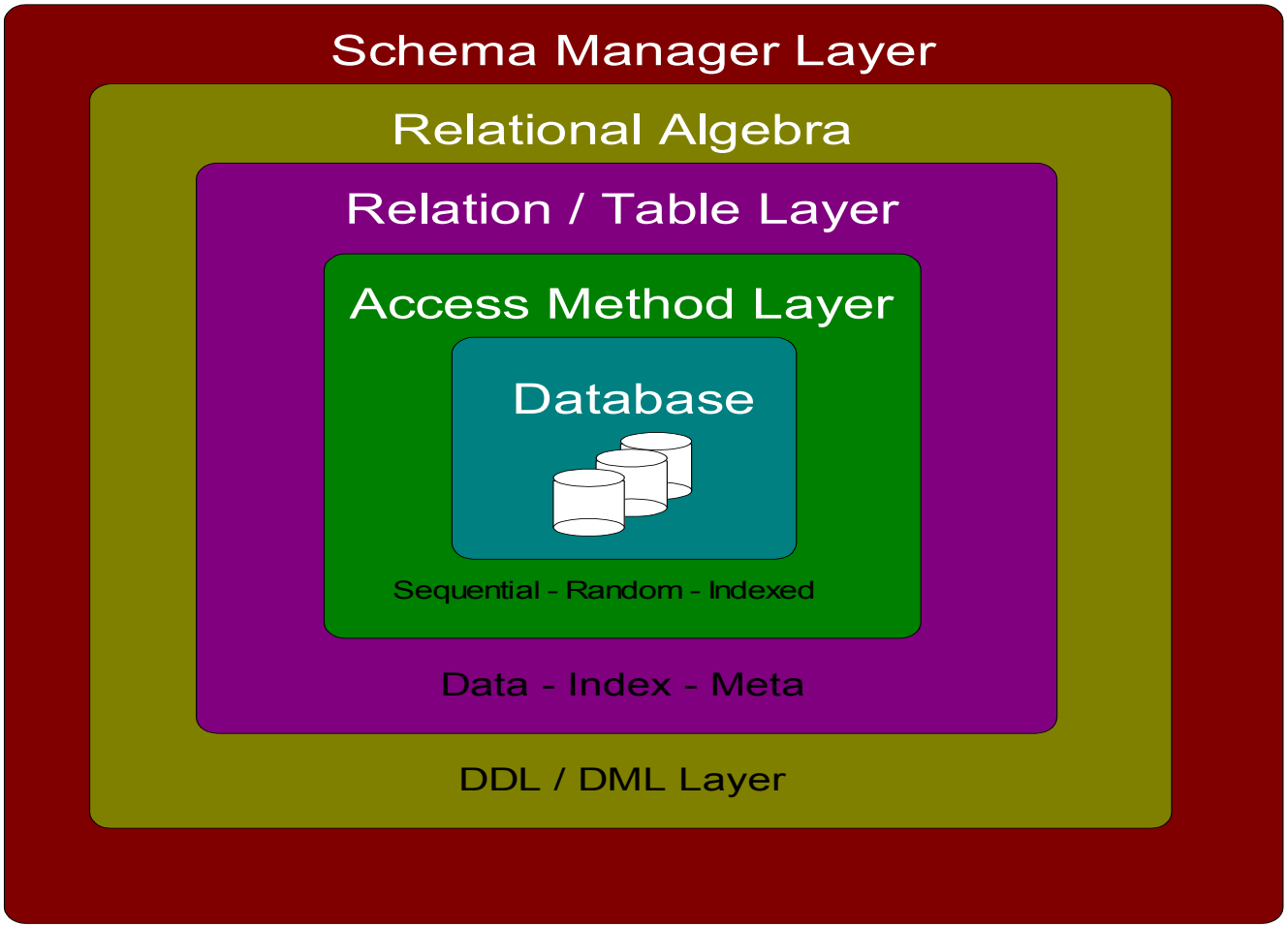
- Design and Development of Mini-DB
- http://www.cs.iusb.edu/minidb/

Objective:

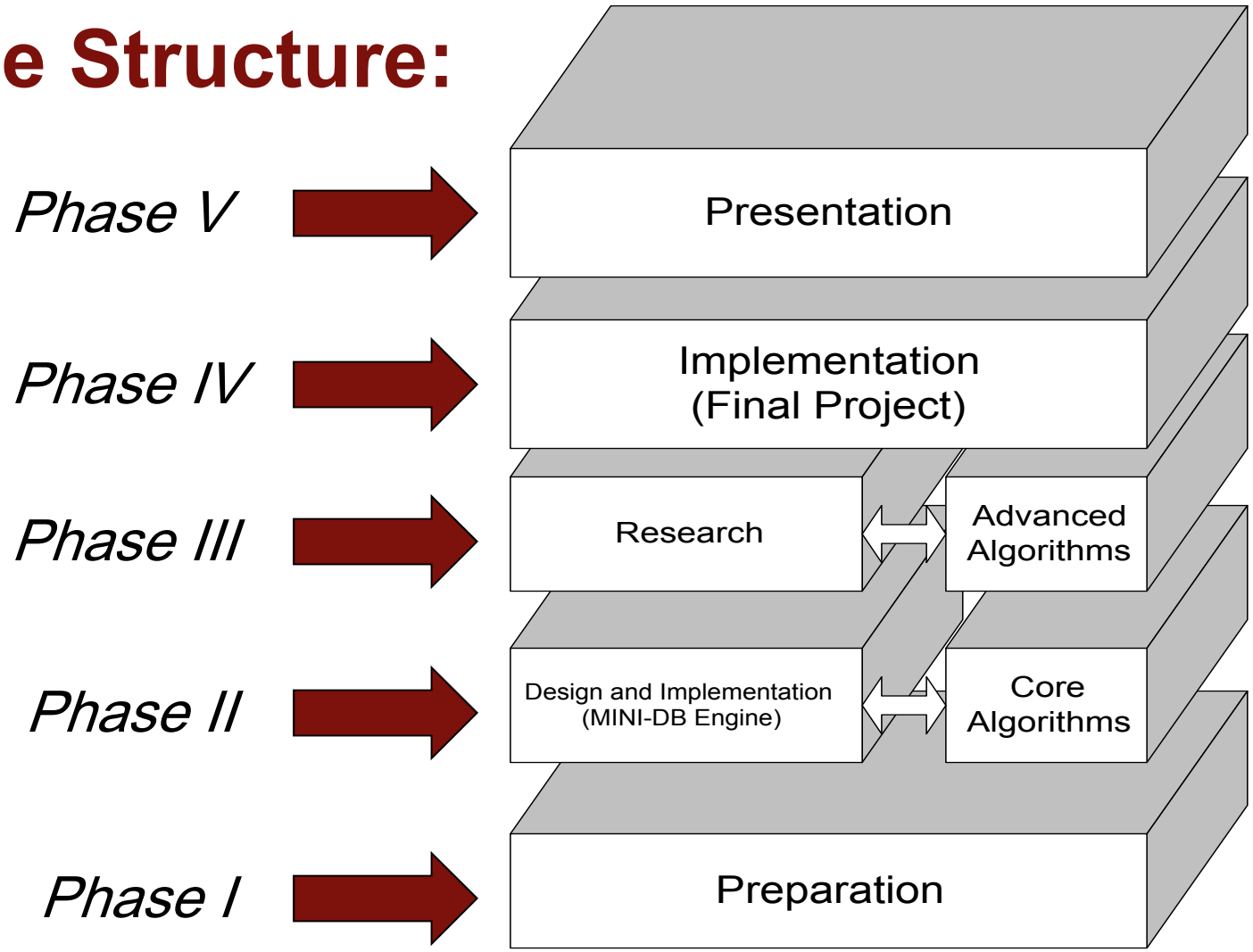
- To Demystify the Inner Workings of Database Systems



MiniDB Conceptual Model:



Course Structure:



Phase I Preparation

Depending on the focus of the course:

Advanced Database Systems

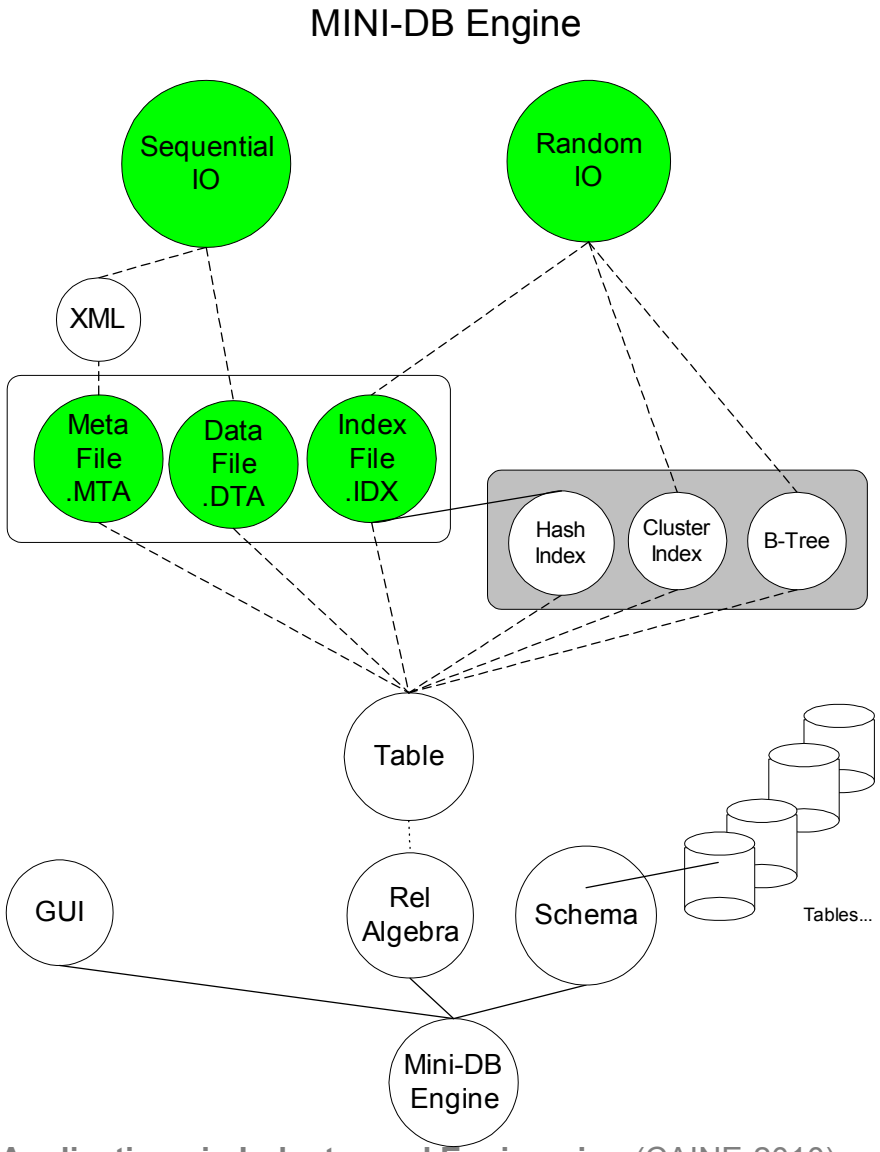
students review and examine the code base for Phase II. (next set of slides)

Database Internals

students survey the I/O facilities of the implementation language. (C++, C, C#, Java, Ruby, etc.)

Phase 2
MiniDB Design

MiniDB Foundation Classes



Phase 2
MiniDB Design

Table Class

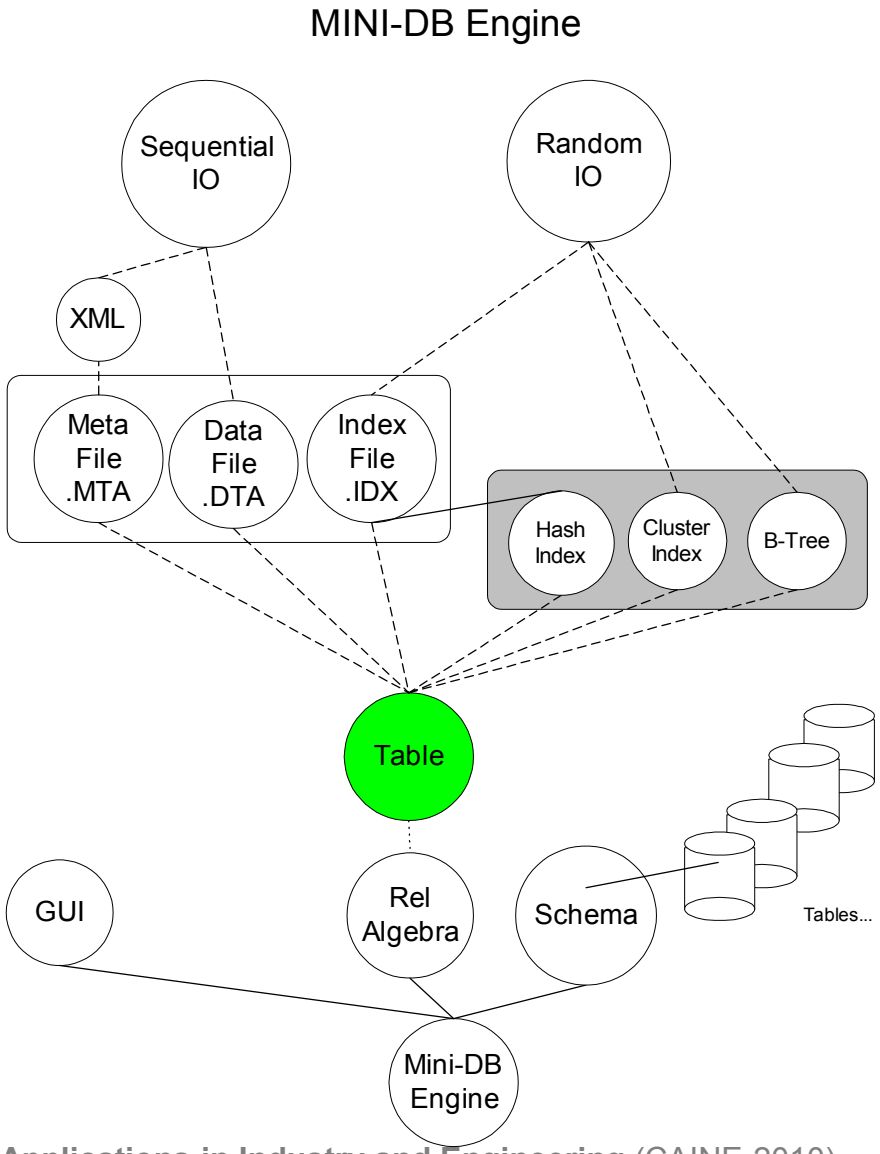
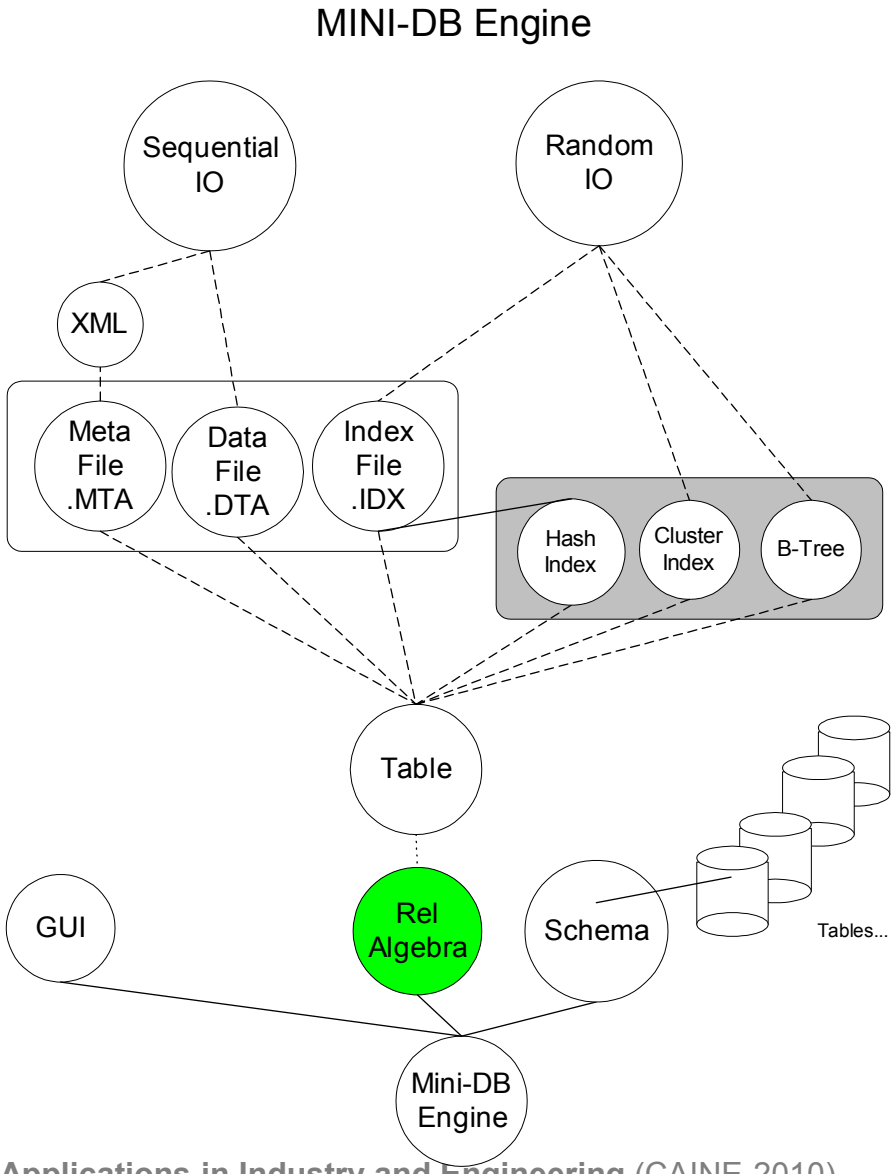


Table Class

```
class Table
{
        char      TableName[256];
        Data_File  *dta;
        Meta_File  *mta;
        Index_File *idx;

        int       TotalRecords;
        int       DeletedRecords;
    public:
        Table(char *tablename);
        ~Table();

        void EraseTable(void);
        int  CreateTable(char *schema);
        void OpenTable(void);
        void CloseTable(void);

        int  Insert(char *a_record, unsigned long key);
        int  Delete(unsigned long key);
        int  Update(char *a_new_record, unsigned long key);

        int  SearchByKey(unsigned long key);
        int  SearchByField(char *field_name, char *value);

        void Print(unsigned long key);
        void PrintSchema(void);
        void Sort();
        void Reorganize();
        int    GetTotalRecords(void);
        int    GetDeletedRecords(void);
        double GarbageRatio(void);
        void   CalculateTotalAndDeletedRecords(void);
};
```

Phase 2
MiniDB Design

Relational Algebra Class

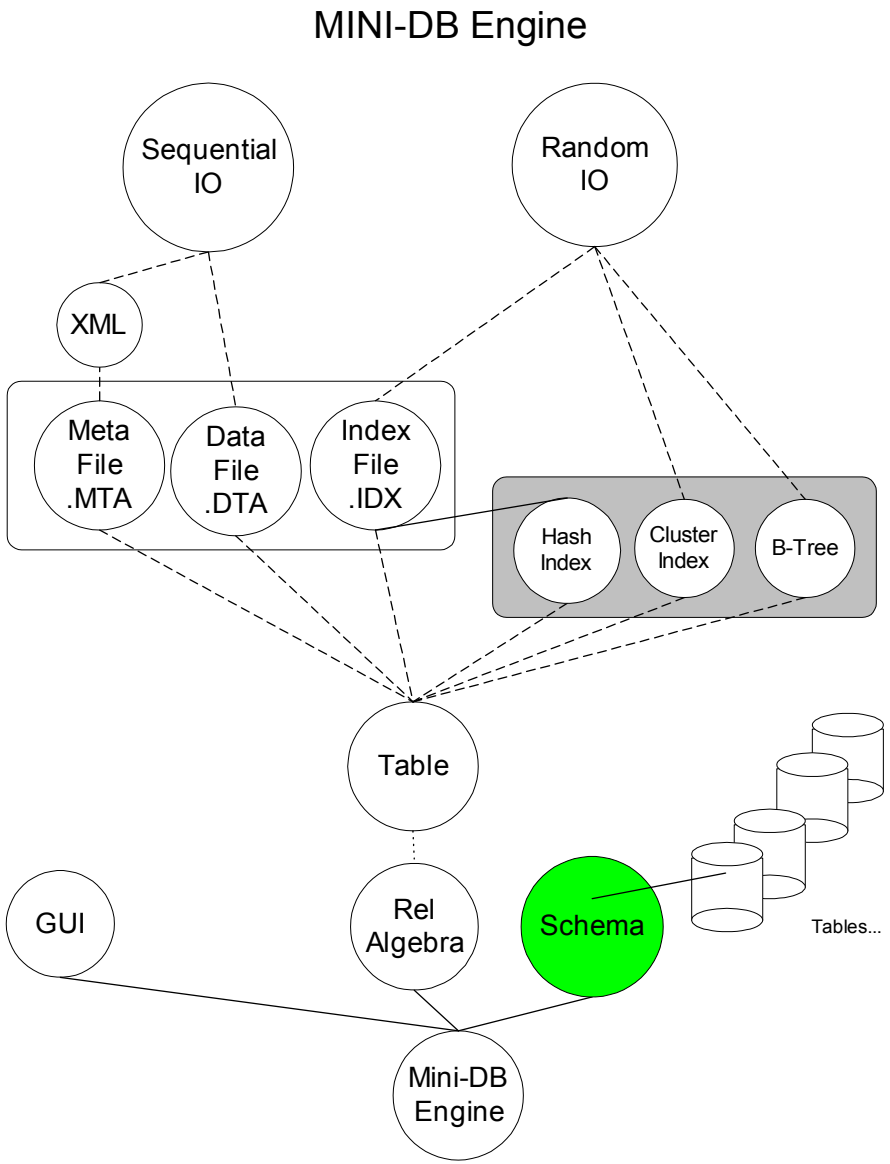


Relational Algebra Class

```
Class Mini_Rel_Algebra {
    bool create(relation_name);
    bool insert(relation_name, attribute_1, value_1,.. attribute_n,
    value_n);
    bool delete(relation_name, attribute_name, attribute_value);
    bool modify(relation_name, attribute_name, attribute_value);
    result_rel select(relation_name, attribute_name, condition,
    attribute_value);
    result_rel project(relation_name, attribute_list);
    result_rel cartesian_product(relation_1, relation_2);
    result_rel join(relation_1, relation_2, condition_list);
    result_rel union(relation_1, relation_2);
    result_rel intersect(relation_1, relation_2);
    result_rel difference(relation_1, relation_2);
}
```

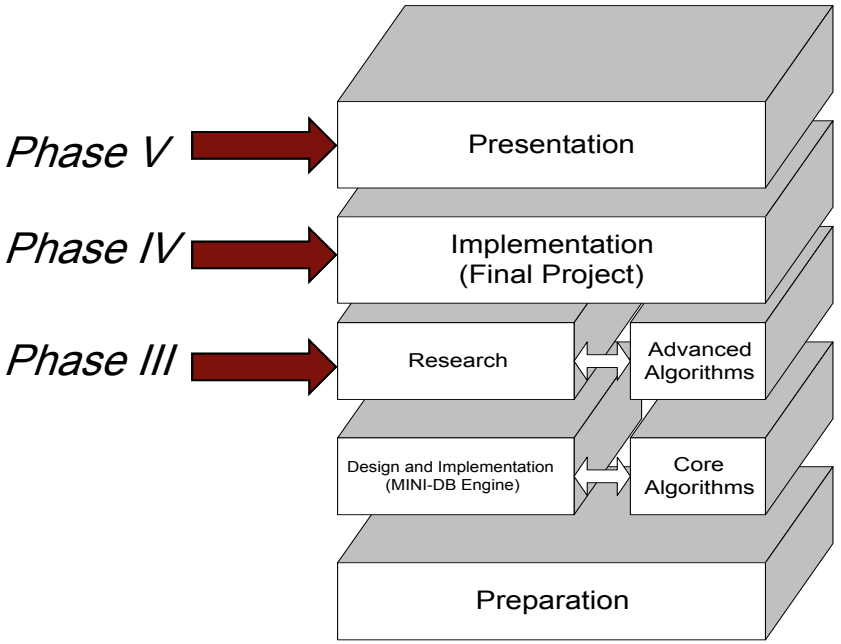
Phase 2
MiniDB Design

Schema Class



Phase 3 Research

- Implementing Phase 1 and 2, may take 6 to 10 weeks, leaving approximately 5 to 9 weeks to work on Phase 3, 4 and 5.



Phase 3

- Phase 3, can be implemented in two ways:

1. A course in Database Internals.

2. A course in Advanced Database Concepts.

Phase 3

Database Internals:

Faculty teaching database internals *can continue to build additional components to extend the MiniDB engine and incorporate features such as:*

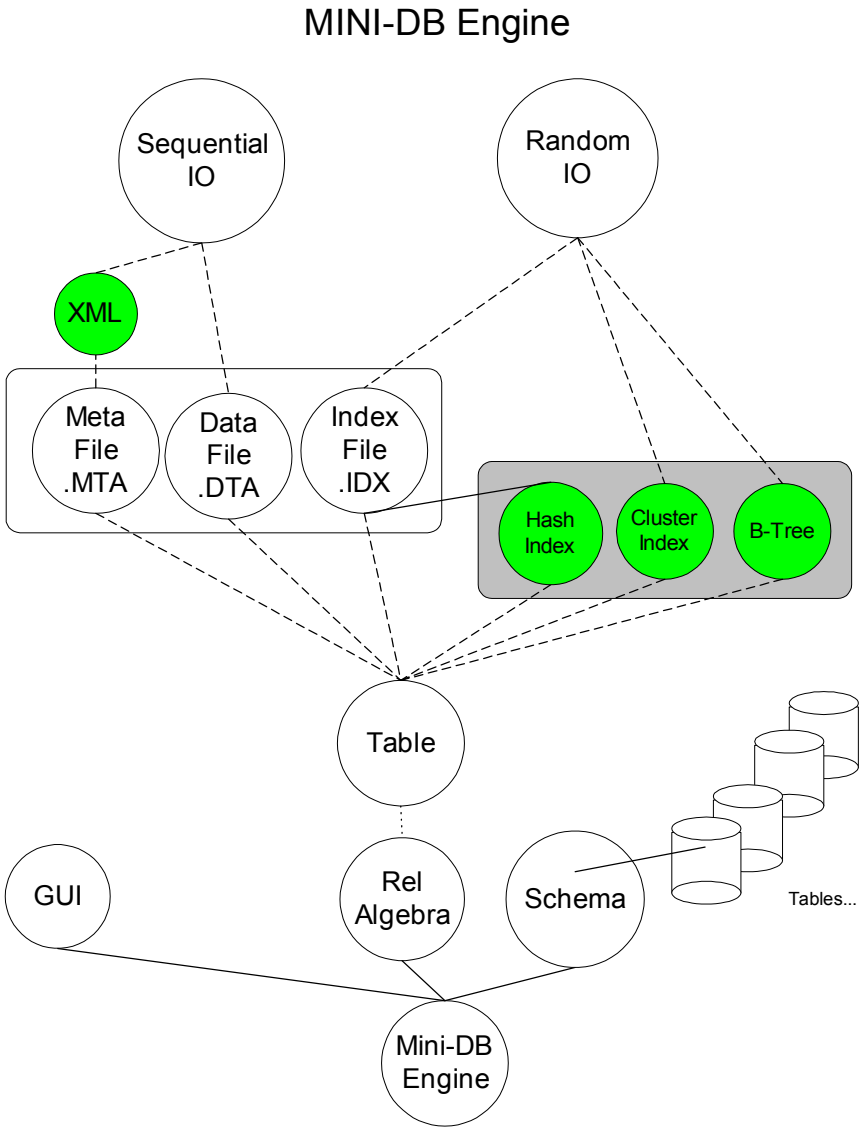
- ***Indexing algorithms*** *(Hash Index, Cluster Index, etc.)*
- ***XML***
- ***Paging and Buffer Management***
- ***Parsing*** *(Relational Algebra and/or SQL parser)*
- ***Log files***

Phase 3

Database Internals:

Supporting Classes

- *Hash index*
- *Cluster index*
- *XML*
- *B-tree*
- *Paging*
- *Caching*
- *SQL Parser*
- *Logging*



Phase 3

Advanced Algorithms:

Faculty teaching advanced database concepts *can start by quickly familiarizing their students with the MiniDB Foundation Classes by way of an assignment (that uses the MFC to build a simple database and then queries the database using the relational algebra API).*

Future assignment can extend the MiniDB engine to incorporate features such as:

- ***Transactions*** *(Start, Commit, Abort, Undo, Redo, checkpoint, write, read)*
- ***Concurrency Control*** *(2PL, Optimistic)*
- ***Distributed Transaction Processing*** *(Implement a new networking class, and extend the MiniDB engine to accommodate distributed query processing)*
- ***Query optimization*** *(Extend the MiniDB engine to include more meta-data as well as runtime information and optimizes the query tree.)*
- ***New and Novel Algorithm*** *(Use the MiniDB platform to implement and compare new algorithms vs. traditional/existing algorithms.*

Lessons Learned:

- *During the past 3 offering of this class, student feedback indicate that after completing this class, they had found a great appreciation for project based classes.*

- *The ability to construct a database engine from scratch was specially appealing. Although, among the students who dropped the course, this aspect of the course was sited as the primary reason.*

- *Students use the code base (MiniDB Foundation Classes) developed in this course in other courses (e.g. Information Organization, and Operating Systems.) as well as after graduation.*

Lessons Learned:

Advanced Database Systems (MiniDB)

- *"**MINI-DB: Demystifying the Inner Workings of Database Systems**", Conference Proceedings of the ISCA 23rd International Conference on Computer Applications in Industry and Engineering (CAINE-2010), Las Vegas, Nevada, November 8 - 10, 2010*

- ***System Development: A Project Based Approach**, ACM-SIGCSE 2009 Conference, Chattanooga, Tennessee, March 4-7, 2009*

Operating Systems (ULTIMA)

- ***"ULTIMA - A Pedagogical Tool for Teaching Operating Systems "**, E-Proceedings of the MICS-2000 Conference, Minneapolis, MN, April 13-15, 2000.*

Computer networks (NetApp - Mini Network API)

- ***NetApp - A Client / Server Applications Builder**, Conference Proceeding of the Small College Computing Symposium (SCCS 98), Fargo, ND, April, 1998.*

Conclusion:

- *We profiled the implementation of a course in "Advanced Database Systems". The primary focus of this course was to study the inner workings of database management systems and to research advanced database concepts.*
- *The course systematically lead the students through the design and implementation of a database engine called MiniDB, then it allowed them to research advanced DB concepts and implement these concepts as part of the MiniDB system.*
- *This approach has allowed our students to use the MiniDB engine as the starting point for further research.*
- *The course material and the MiniDB project is available as an open courseware.*

Interested?

The MiniDB is available as an open source courseware:

- *www.cs.iusb.edu/minidb*

The site includes:

- *Assignments*
- *Design Documentation*
- *C++ API*
- *Source Code (Restricted Distribution to Faculty only)*

Other MiniDB Projects:

Project:	*Minibase (Inspired by Minirel)*
Author:	*Mike Carey and Raghu Ramakrishnan (Univ. of Wisconsin)*
Language:	*C++*
URL	*http://pages.cs.wisc.edu/~dbbook/openAccess/Minibase/minibase.html*
Status	*Active*
Project:	*Minirel*
Author:	*David DeWitt*
Language:	*C*
URL	*Not available*
Status	*May be inactive*
Project:	*SimpleDB*
Author:	*Edward Sciore (Boston College)*
Language:	*Java*
URL	*http://cs.bc.edu/~sciore/simpledb/intro.html*
Status	*Active*
Project:	*MinSQL*
Author:	
Language:	*Java*
URL	
Status	*Not open source*
Project:	*miniDB*
Author:	*Hans Harder*
Language:	*C*
URL	*http://freshmeat.net/projects/minidb/* *http://www.atbas.org/minidb/index.php*
Status	*Active*
Project:	*minidb*
Author:	*jpwarren00*
Language:	*Java*
URL	*http://code.google.com/p/minidb/*
Status	*May be inactive*